

FIG. 1

VIDEO FILE SERVER

```

graph TD
    Start([CLIENT SERVING]) --> 50{REALTIME STREAMING?}
    50 -- YES --> 55{NETWORK CONGESTION?}
    50 -- NO --> 51{INPUT NEW MPEG-2 FILE?}
    51 -- YES --> 52[INPUT NEW MPEG-2 FILE AND CREATE REDUCED-QUALITY MPEG-2 FILE AS AVAILABLE RESOURCES PERMIT]
    51 -- NO --> 53{PLAY LIST EDITING?}
    53 -- YES --> 54[BROWSE THROUGH REDUCED-QUALITY MPEG-2 FILE TO SELECT IN-POINTS AND OUT-POINTS OF CLIPS TO BE SPLICED]
    53 -- NO --> 50
    52 --> 56[STREAM COMPRESSED VIDEO FROM REDUCED-QUALITY MPEG-2 FILE]
    54 --> 56
    55 -- YES --> 56
    55 -- NO --> 57{REDUCED QUALITY REQUESTED?}
    57 -- YES --> 56
    57 -- NO --> 58{TRICK MODE REQUESTED?}
    58 -- YES --> 59{LOW SPEED-UP?}
    59 -- YES --> 60[STREAM ORIGINAL QUALITY I-FRAMES AND 3 FREEZE FRAMES PER I-FRAME]
    59 -- NO --> 61[SELECT 1 OR 2 FREEZE FRAMES PER I-FRAME FOR DESIRED SPEED-UP]
    61 --> 62[STREAM REDUCED-QUALITY I-FRAMES AND INSERTED FREEZE FRAMES]
    58 -- NO --> 63[STREAM ORIGINAL QUALITY MPEG-2 CODED VIDEO]
    60 --> End([END])
    62 --> End
    63 --> End

```

FIG. 2

0 **1** **2** **3** **4** **5** **6** **7** **8** **9**



124

000000-61680960

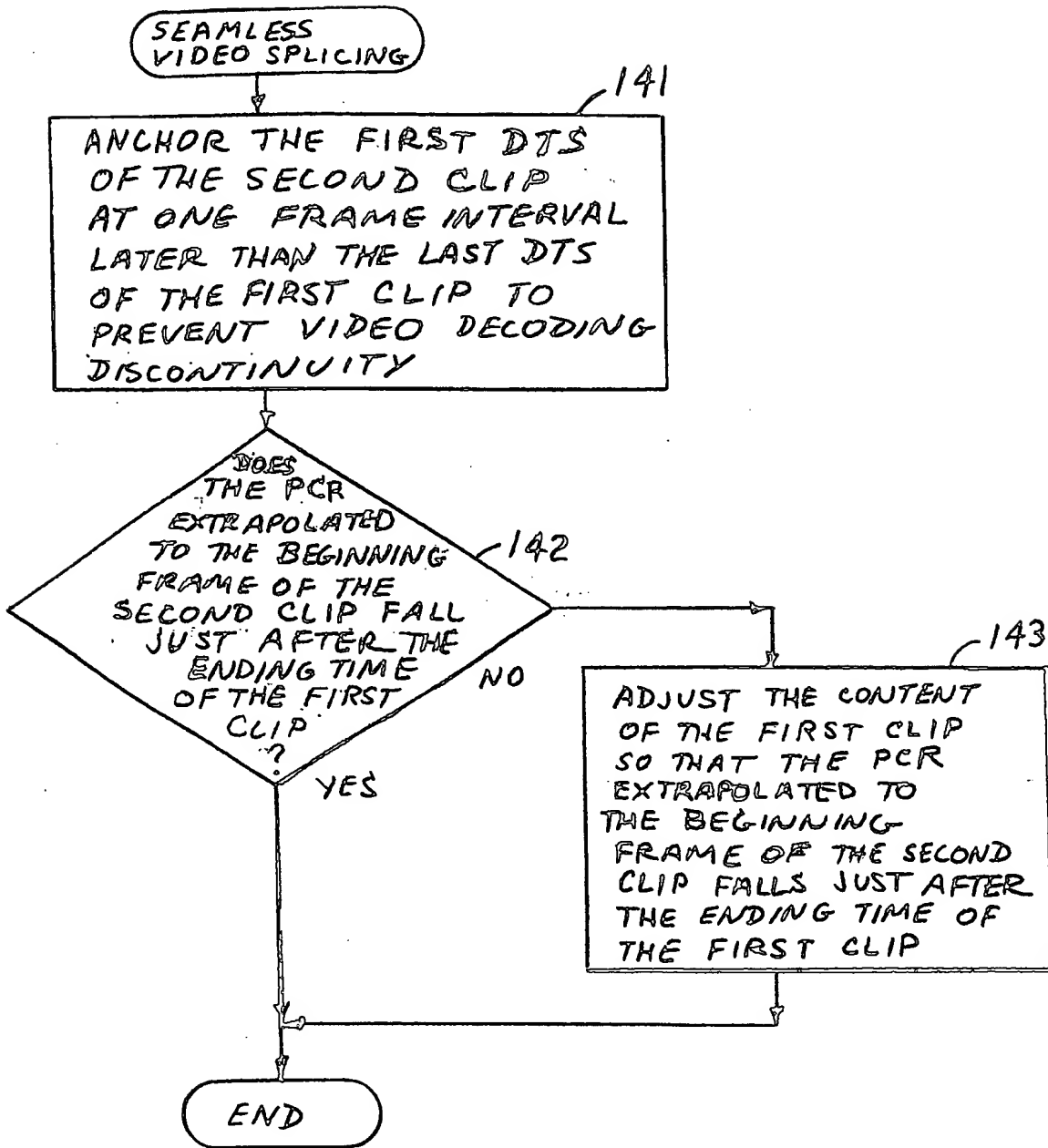


FIG. 4

VIDEO
SPLICING

DETERMINE THE LAST DTS/PTS
OF THE FIRST CLIP
(DTS_{L1})

DETERMINE THE TIME OF ARRIVAL (T_e) OF THE LAST BYTE OF THE FIRST CLIP

ADD ONE FRAME INTERVAL
TO DTS_{H1} TO FIND THE
DESIRED FIRST DTS LOCATION
FOR THE SECOND CLIP
($DTS_{F1} = DTS_{H1} + 1/FR$)

KEEPING THE DTS-PCR_e
RELATION UNALTERED FOR
THE SECOND CLIP, FIND THE
TIME INSTANT T_s AT WHICH
THE FIRST BYTE OF THE
SECOND CLIP SHOULD
ARRIVE

$$(T_{\text{START}} = \text{DTS}_{F2} - \text{PCRe}_2)$$
$$(T_S = DT_{SF1} - T_{START})$$

B

FIG. 5

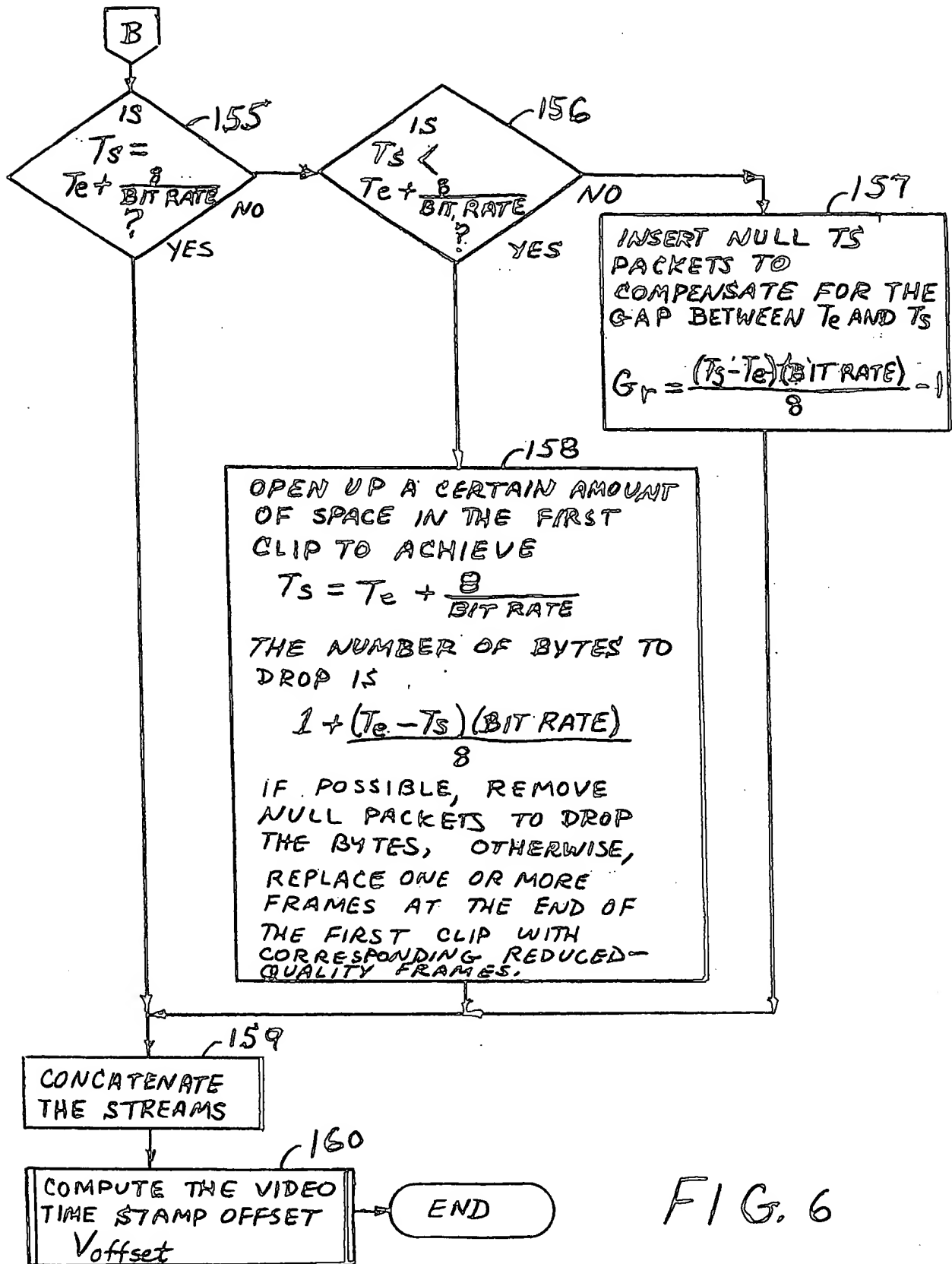


FIG. 6


```

graph TD
    Start([TRICK MODE  
STREAM]) --> 181[Input MPEG-2 TS from which a  
trick mode clip will be extracted.]
    181 --> 182[Video elementary stream (VES)  
extracted.]
    181 --> 183[Audio elementary stream (AES)  
extracted.]
    182 --> 184[I frame extraction and valid PES  
formation.]
    184 --> 185[SNR scaling of the I-frames-only PES]
    185 --> 186[Freeze P frame insertion and valid PES  
formation.]
    183 --> 187[Selection and concatenation of the  
appropriate audio access units (from  
the original asset) based on the  
structure of the VES in the trick mode  
clip and valid PES encapsulation  
around these audio access units.]
    186 --> 188[TS stream generation by multiplexing the  
above video PES into a system info (SI)  
and audio PES carrying TS skeleton.]
    187 --> 188
    188 --> End([END])

```

FIG. 10

FIG. 10

FIG. 11
(PRIOR ART)

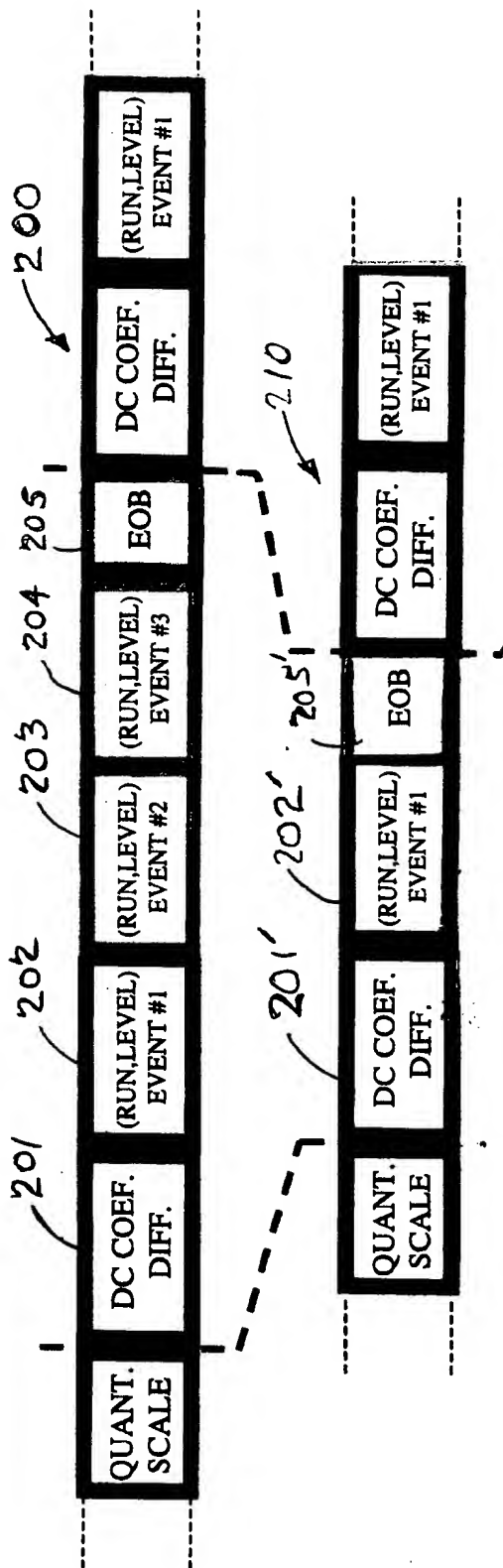


FIG. 12

FIG. 13

(FDSNR-LP

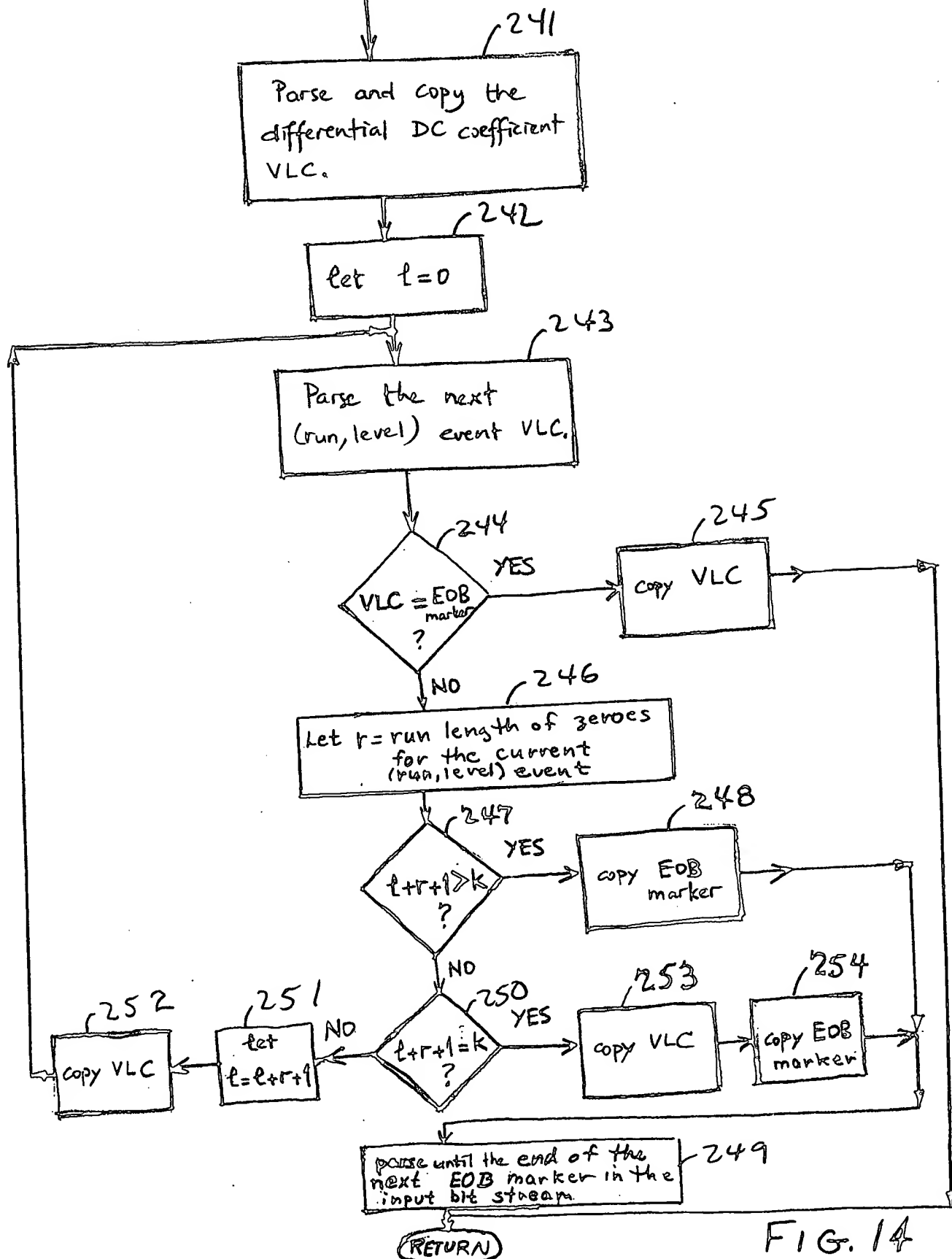


FIG. 14

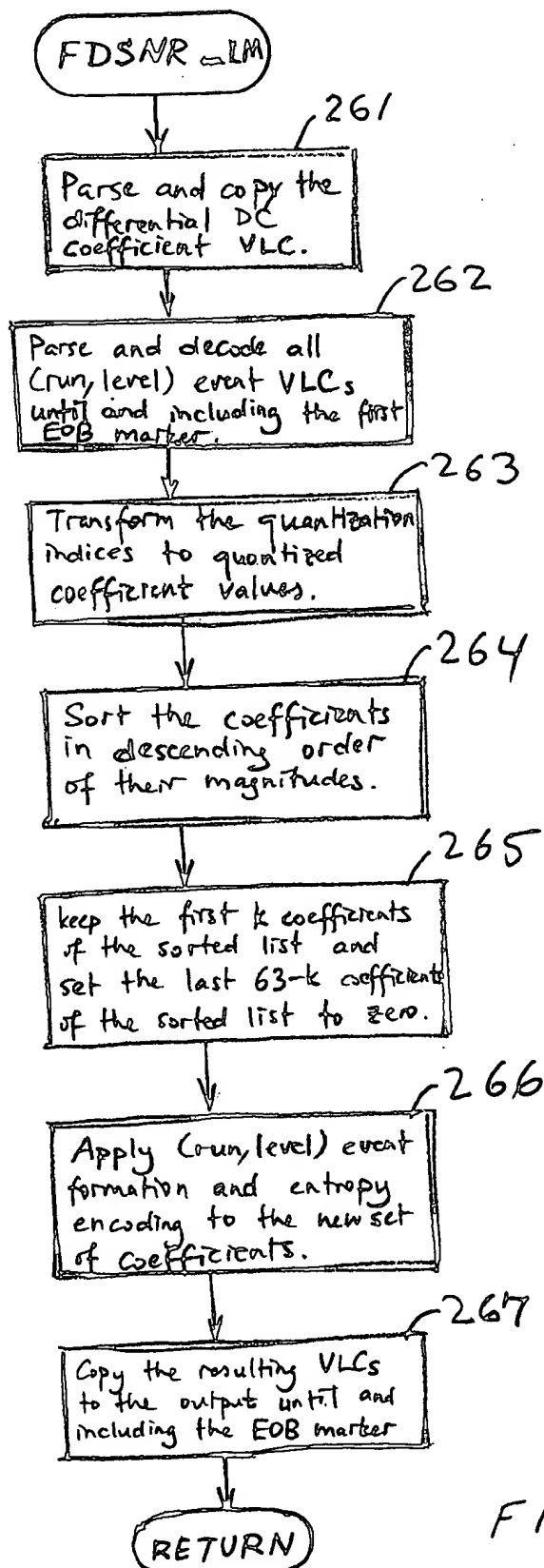
[illegible]

FIG. 15

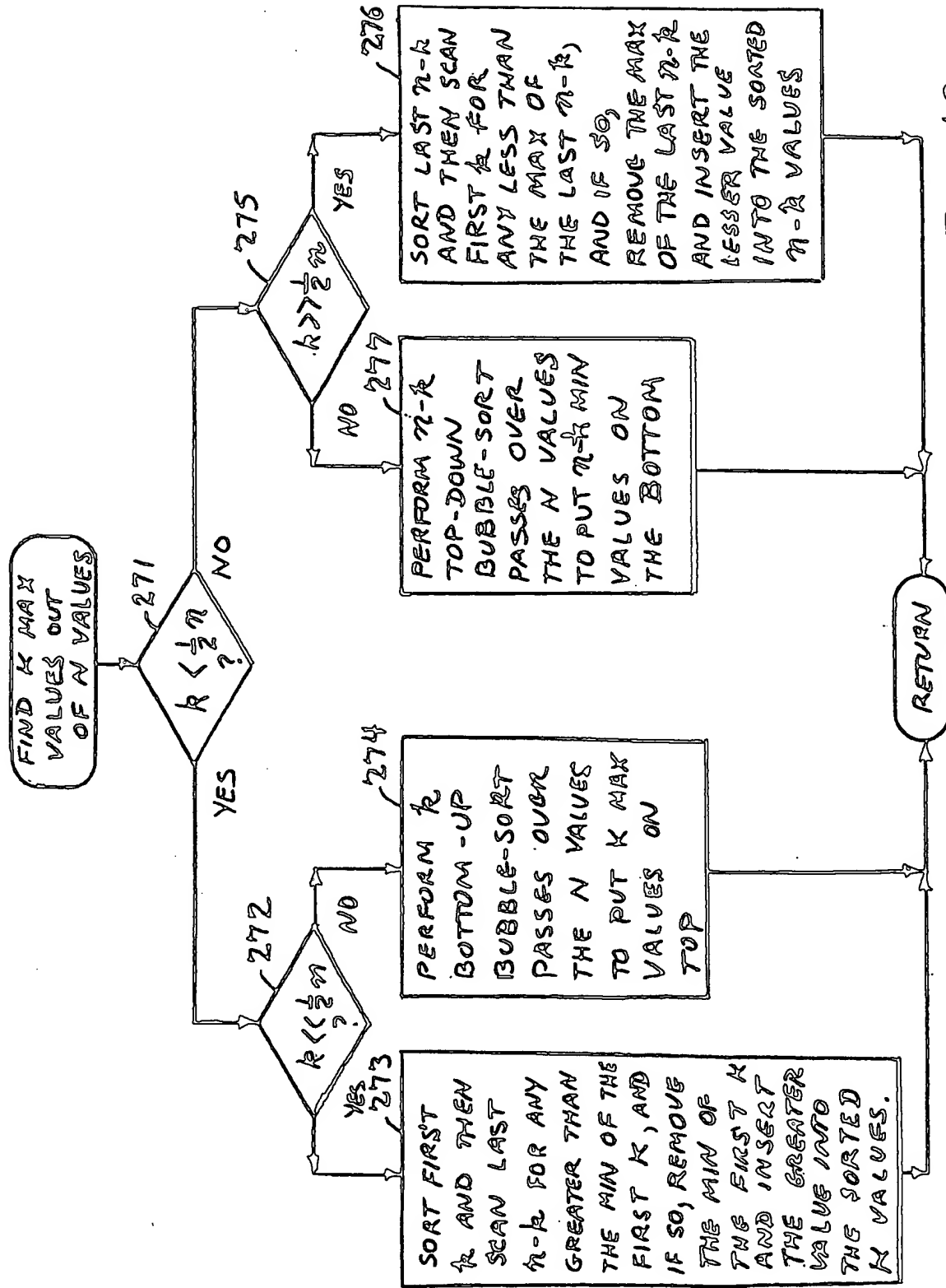


FIG. 16

```

graph TD
    Start(( )) --> 280([SORT K FROM N])
    280 --> 281[i ← 0]
    281 --> 282[GET NEXT COEFFICIENT FROM INPUT STREAM]
    282 --> 283{EOB?}
    283 -- YES --> 287([RETURN])
    283 -- NO --> 284{i ≤ K?}
    284 -- YES --> 285[PUT COEFFICIENT INDEX AND MAGNITUDE INTO SORT LIST]
    285 --> 286[i ← i + 1]
    286 --> 282
    284 -- NO --> 287
    286 --> 287
    287 --> 288([C])
    288 --> 290{COEFF. MAGNITUDE > MAGNITUDE AT END OF LIST?}
    290 -- YES --> 291[REMOVE ENTRY AT THE END OF THE LIST]
    290 -- NO --> 292[GET NEXT COEFFICIENT FROM INPUT STREAM]
    292 --> 293{EOB?}
    293 -- YES --> 294([RETURN])
    293 -- NO --> 295([C])
    295 --> 296[INSERT CURRENT COEFFICIENT INDEX AND MAGNITUDE INTO THE LIST AT THE RANK POSITION]
    296 --> 297([C])
    297 --> 290
    291 --> 290
    
```

④

[illegible]

FIG. 18.


```

graph TD
    Start([APPROXIMATE SORT K FROM N]) -- 311 --> Clear[311 CLEAR HASH TABLE]
    Clear --> GetNext[312 GET NEXT COEFFICIENT FROM INPUT STREAM]
    GetNext --> EOB{313 EOB ?}
    EOB -- YES --> Strip[314 STRIP HASH TABLE INDEX FROM MSBs OF COEFFICIENT MAGNITUDE]
    EOB -- NO --> Strip
    Strip --> Insert[315 INSERT COEFFICIENT INDEX ON HASH LIST OF INDEXED HASH TABLE ENTRY]
    Insert --> GetNext
    Insert --> Index[316 i ← 2^m - 1  
j ← k]
    Index --> IndexHash[317 INDEX HASH TABLE WITH i]
    IndexHash --> EntryEq0{318 ENTRY = 0 ?}
    EntryEq0 -- YES --> DecI[320 i ← i - 1]
    EntryEq0 -- NO --> GetNextEntry[321 GET NEXT ENTRY FROM HASH LIST AND PUT COEFFICIENT IN THE OUTPUT STREAM]
    DecI --> I0{319 i = 0 ?}
    I0 -- YES --> Return1([RETURN])
    I0 -- NO --> DecI
    GetNextEntry --> EndList{322 END OF LIST ?}
    EndList -- YES --> DecJ[323 J ← J - 1]
    EndList -- NO --> GetNextEntry
    DecJ --> JLe0{324 J ≤ 0 ?}
    JLe0 -- YES --> Return2([RETURN])
    JLe0 -- NO --> GetNextEntry
  
```

The flowchart illustrates an approximate sorting algorithm. It begins with an initial step to approximate sort K from N, leading to clearing the hash table (311). The process then enters a loop where it gets the next coefficient from the input stream (312). If the end of the input stream (EOB) is reached (313), it proceeds to strip the hash table index from the MSBs of the coefficient magnitude (314) and insert the coefficient index on the hash list of the indexed hash table entry (315). If EOB is not reached, it proceeds to strip the hash table index from the MSBs of the coefficient magnitude (314) and insert the coefficient index on the hash list of the indexed hash table entry (315). The process then initializes i to 2^m - 1 and j to k (316). It then enters a loop where it indexes the hash table with i (317). If the entry is equal to 0 (318), it decrements i (320). If i is 0 (319), it returns. If the entry is not 0, it gets the next entry from the hash list and puts the coefficient in the output stream (321). It then checks if it is the end of the list (322). If yes, it decrements j (323). If j is less than or equal to 0 (324), it returns. If not, it continues the loop.

FIG. 19

MODIFIED
FDSNR - LM

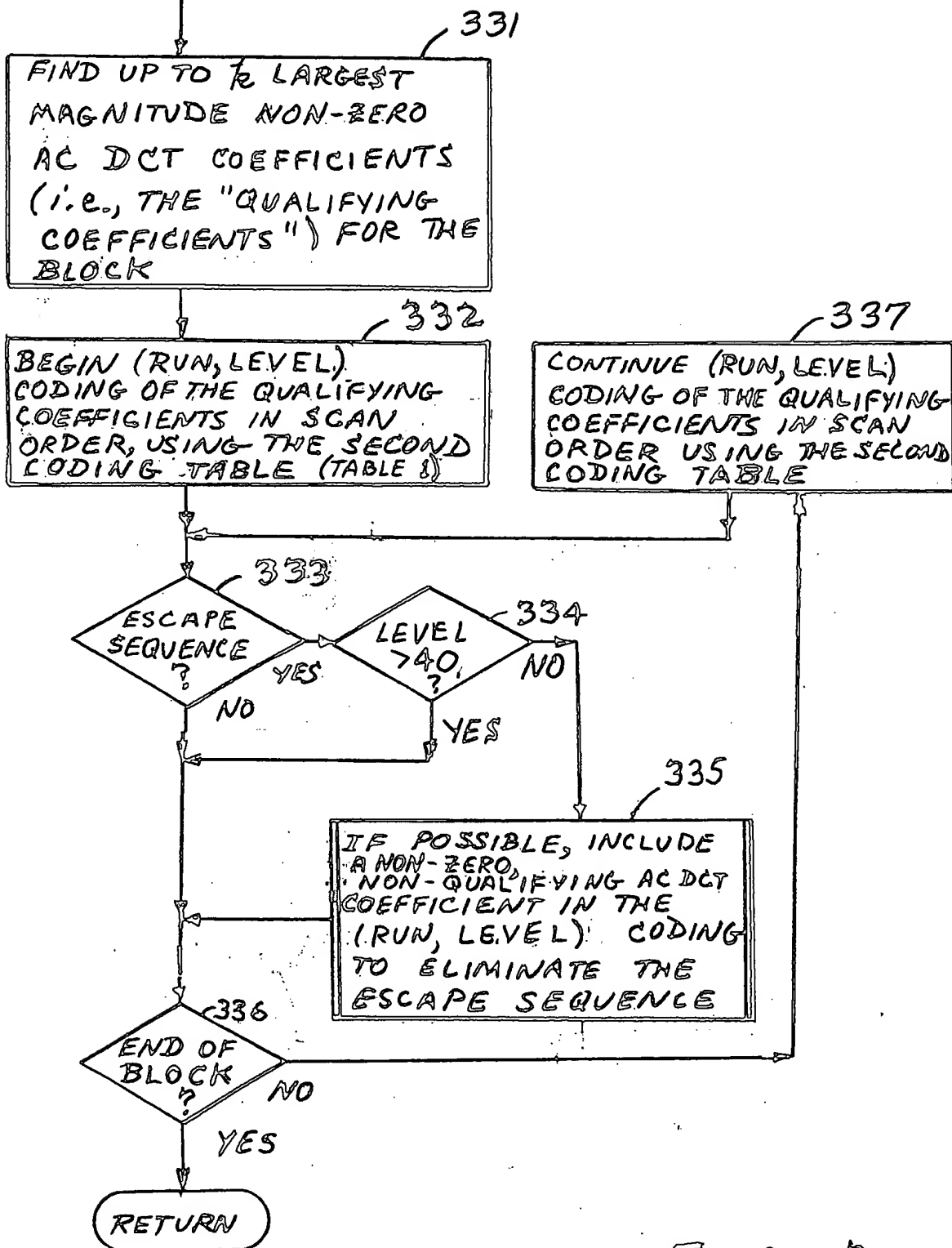


FIG. 20

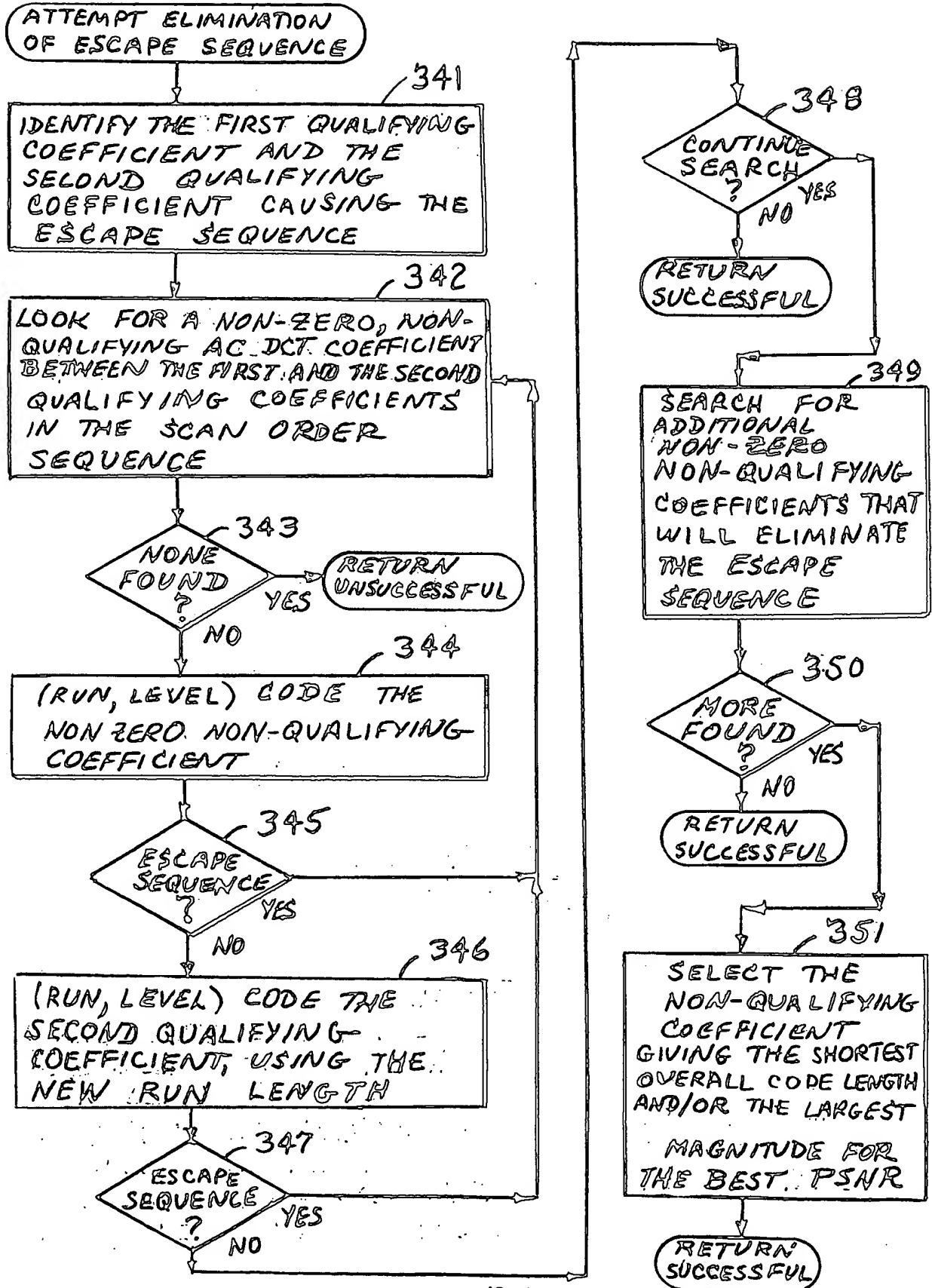


FIG. 21

```

graph TD
    Start([I]) --> 361[MPEG SCALING  
k ← 9, QSF ← 2]
    361 --> 362{PICTURE  
HEADER?}
    362 -- NO --> 384{END OF  
CLIP?}
    362 -- YES --> 363{intra-  
vlc-format  
= 0?}
    363 -- YES --> 364[READ IN  
TABLE 0]
    363 -- NO --> 365[READ IN  
TABLE 1]
    364 --> 366[APPLY MODIFIED FDSNR-LM  
PROCEDURE, USING ADJUSTED  
QUANTIZER SCALE  
INDEX IF LESS THAN THE  
MAXIMUM POSSIBLE  
QUANTIZER SCALE INDEX]
    365 --> 366
    366 --> 367{SLICE  
HEADER?}
    367 -- NO --> 384
    367 -- YES --> 368{ESCAPE  
FREQUENCY  
≥ TH1?}
    368 -- YES --> 369{QSF ≤ 2?}
    368 -- NO --> F([F])
    369 -- YES --> 370[QSF ← QSF * 2]
    369 -- NO --> G([G])
    370 --> H([H])
    384 -- YES --> RETURN([RETURN])
    384 -- NO --> 362

```

FIG. 22

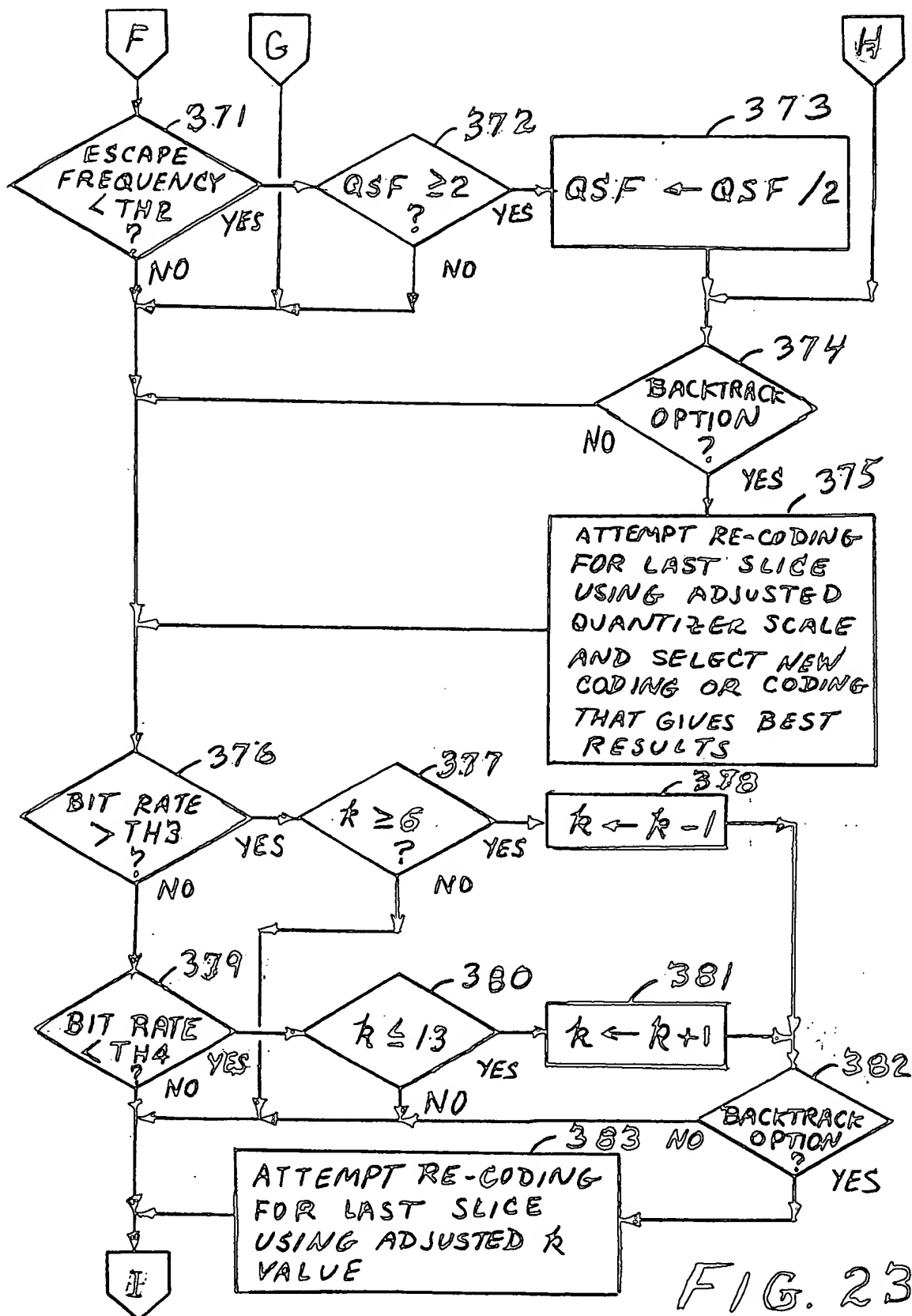


FIG. 23

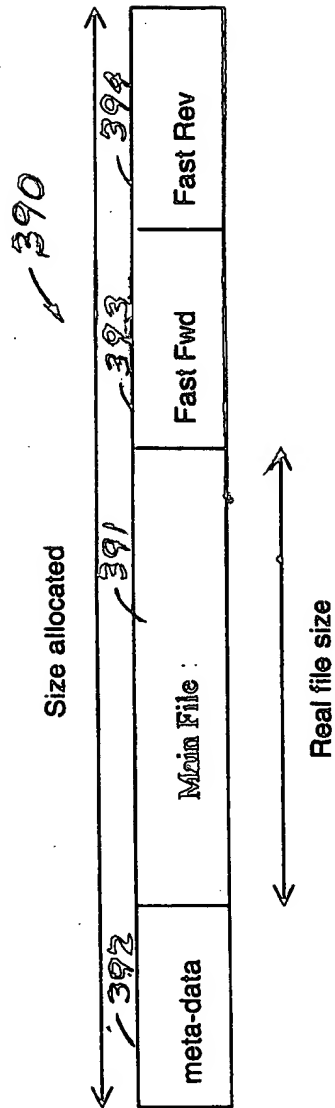
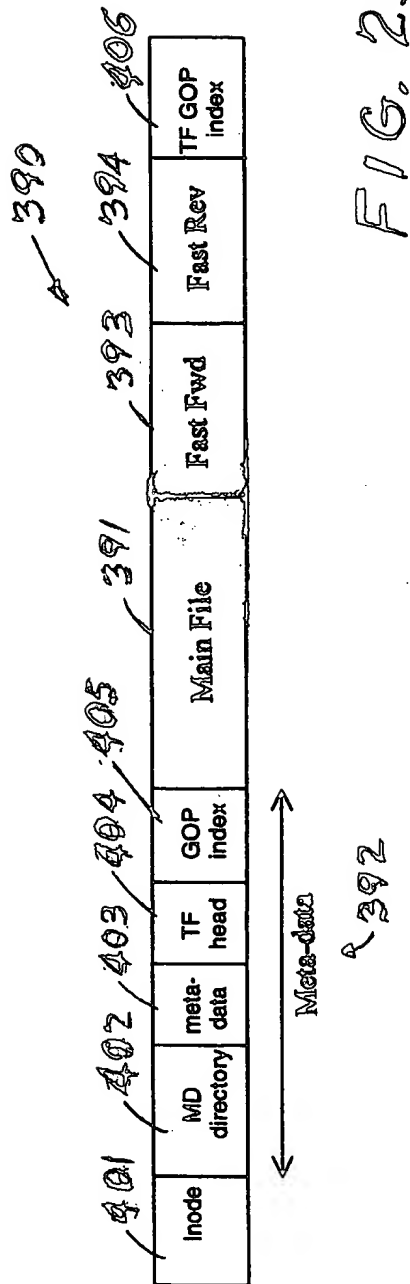


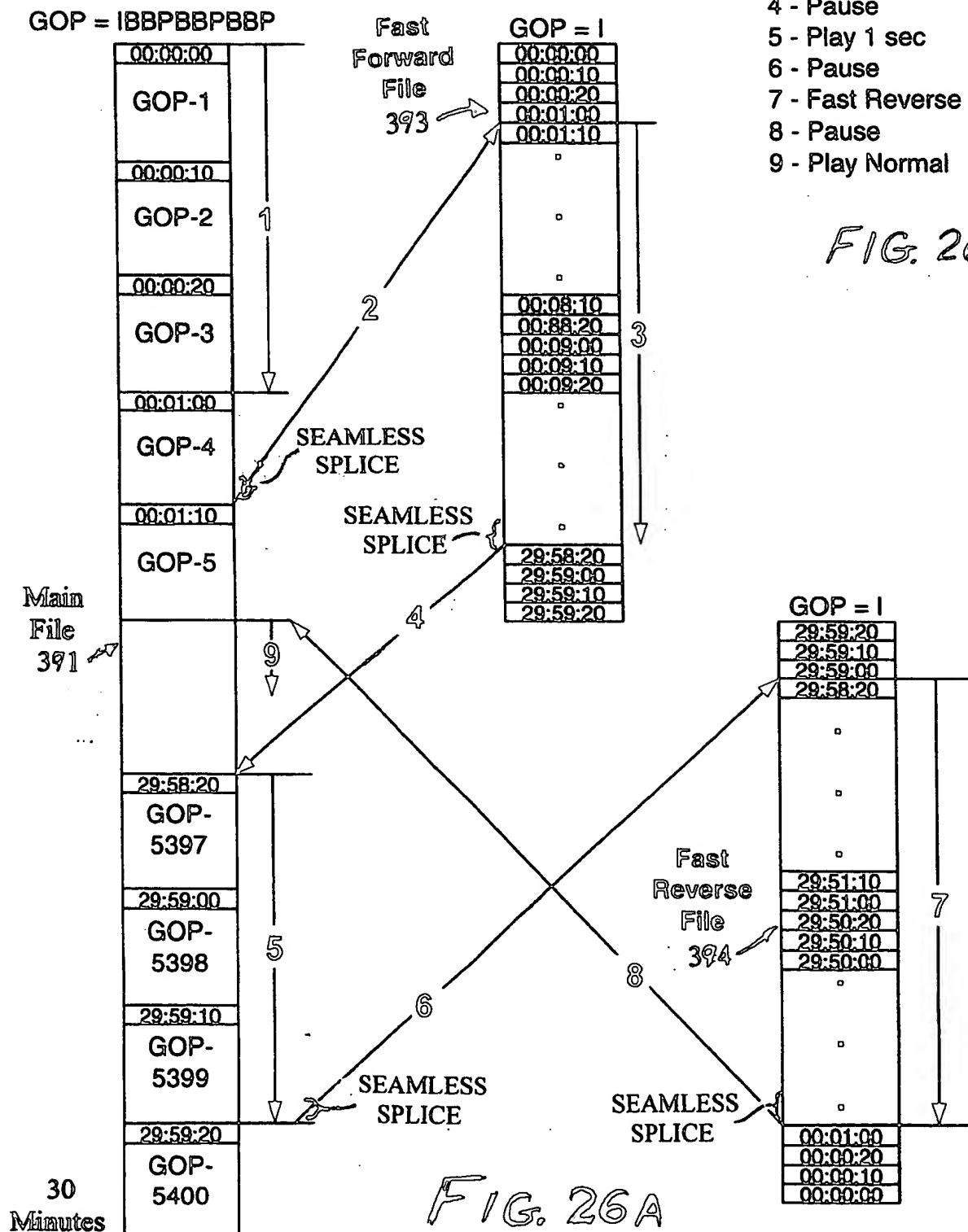
FIG. 24

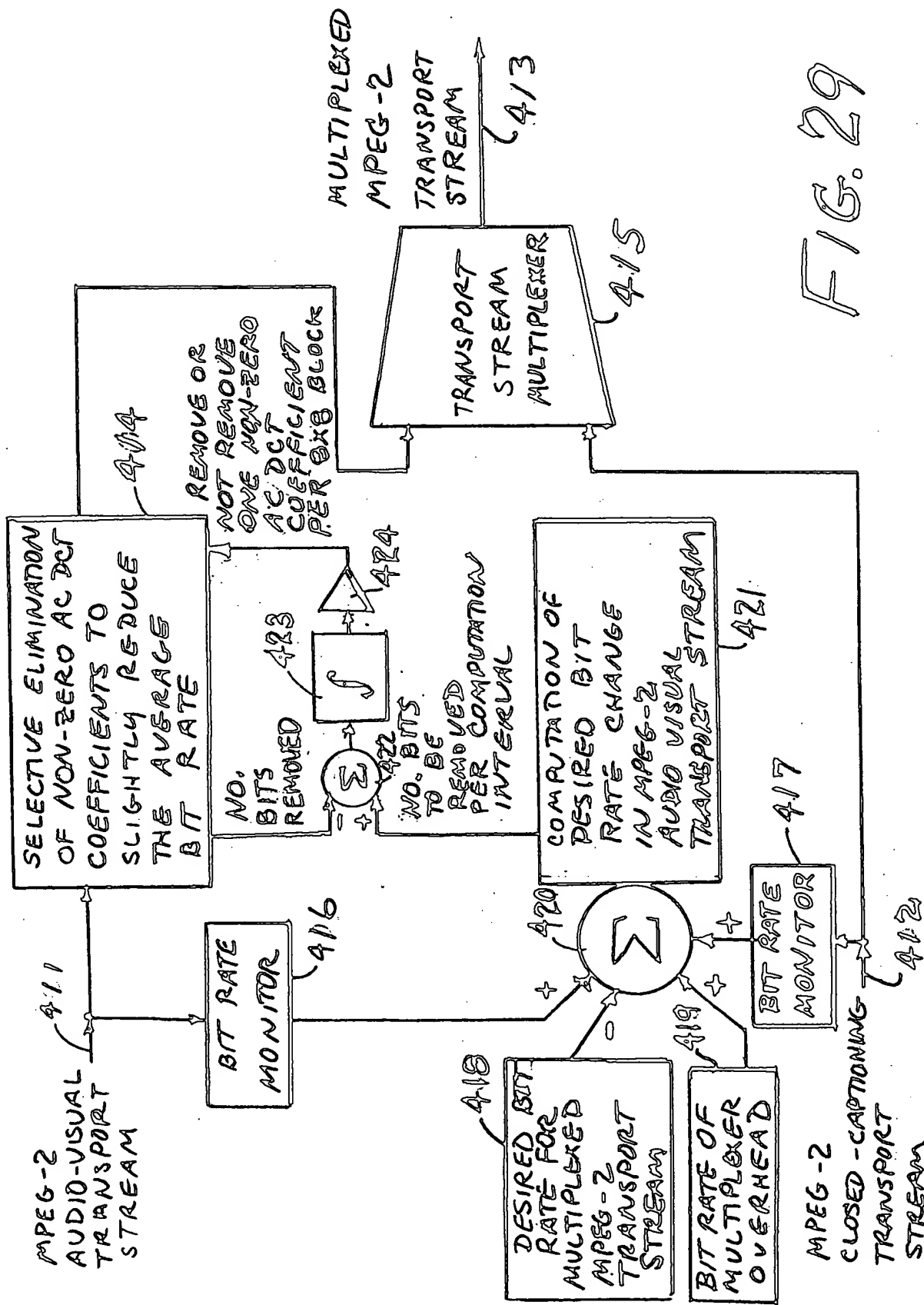


THE UNIVERSITY OF CHICAGO

- 1 - Play from start 1 sec
- 2 - Pause
- 3 - Fast Forward to 29 min
- 4 - Pause
- 5 - Play 1 sec
- 6 - Pause
- 7 - Fast Reverse to 1 sec
- 8 - Pause
- 9 - Play Normal

FIG. 26B





261